Amendments to the Claims:

Claim 1. (Currently amended) A computer implemented method for producing a binary-level conditional branch reversal within a binary program on a computer architecture that supports a predicated execution, comprising:

employing runtime data that has been collected on a compiler to determine a binary-level conditional branch to be reversed;

obtaining a predicate expression representing a condition that influences a direction of program flow of the binary-level conditional branch to be reversed;

determining a binary-level transformation based on the predicate expression that causes the binary-level conditional branch to be triggered when an opposite condition is true; and

~~modifying the binary-level conditional branch with the~~ applying the determined binary-level transformation to reverse the binary-level conditional branch~~, wherein the binary-level conditional branch is reversed~~.

Claim 2. (Original) The computer-implemented method of claim 1, wherein obtaining the predicate expression comprises:

uniquely identifying predicates that influence the direction of program flow of the binary-level conditional branch to be reversed;

deducing relationships between the uniquely identified predicates; and

based on the relationships between the uniquely identified predicates, determining at least one predicate that influences the direction of program flow of the binary-level conditional branch.

Claim 3. (Original) The computer-implemented method of claim 2, further comprising locating speculative load instructions that impacts the truth value of a predicate associated with the binary-level conditional branch.

Claim 4. (Original) The computer-implemented method of claim 3, further comprising inserting into the binary program instructions to exclude execution of the binary-level conditional branch in response to a faulted speculative load.

Claim 5. (Original) The computer-implemented method of claim 2, wherein deducing the relationships includes conducting a predicate-aware, reaching definition data flow analysis.

Claim 6. (Original) The computer implemented method of claim 1, wherein determining the binary-level transformation comprises computing an inverse predicate expression that describes the opposite condition.

Claim 7. (Original) The computer implemented method of claim 6, further comprising determining whether at least one predicate in the inverse predicate expression is unmaterialized, and, if so, materializing the unmaterialized predicate.

Claim 8. (Original) The computer implemented method of claim 7, wherein materializing the unmaterialized predicate comprises:

locating a free register to support the unmaterialized predicate;

associating a new predicate with the free register; and

adding an instruction to define the new predicate as the unmaterialized predicate, wherein the unmaterialized predicate is now materialized.

Claim 9. (Original) The computer implemented method of claim 8, wherein locating the free register comprises conducting a predicate-aware liveness analysis.

Claim 10. (Original) The computer implemented method of claim 6, wherein if the inverse predicate expression includes multiple predicates, reducing the inverse predicate expression to a single predicate.

Claim 11. (Original) The computer implemented method of claim 10, wherein modifying the binary-level conditional branch comprises replacing an existing guarding predicate with the single predicate.

Claims 12 – 21. (Withdrawn)

Claim 22. (Currently amended) A computer-readable medium having computer-executable instructions for producing a binary-level conditional branch reversal within a binary program on a computer architecture that supports a predicated execution, the instructions comprising:

employing runtime data that has been collected on a compiler to determine a binary-level conditional branch to be reversed;

obtaining a predicate expression representing a condition that influences a direction of program flow of the binary-level conditional branch to be reversed;

determining a binary-level transformation based on the predicate expression that causes the binary-level conditional branch to be triggered when an opposite condition is true; and

~~modifying the binary-level~~ conditional ~~branch with the~~ applying the determined binary-level transformation to reverse the binary-level conditional branch~~, wherein the binary-level conditional branch is reversed~~.